Terms used
register and **live** and **physical** and **logical** and **allocat%** and **stack** and **restor%**

Found **15,095** of **155,867**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Speculative execution exception recovery using write-back suppression

Roger A. Bringmann, Scott A. Mahlke, Richard E. Hank, John C. Gyllenhaal, Wen-mei W. Hwu

December 1993 **Proceedings of the 26th annual international symposium on Microarchitecture**

Full text available: pdf(1.22 MB) Additional Information: full citation, references, citings

Keywords: VLIW, exception detection, exception recovery, scheduling, speculative execution, superscalar

2 Sentinel scheduling: a model for compiler-controlled speculative execution

Scott A. Mahlke, William Y. Chen, Roger A. Bringmann, Richard E. Hank, Wen-Mei W. Hwu, B. Ramakrishna Rau, Michael S. Schlansker

November 1993 **ACM Transactions on Computer Systems (TOCS)**, Volume 11 Issue 4

Full text available: pdf(2.26 MB) Additional Information: full citation, abstract, references, citings, index terms

Speculative execution is an important source of parallelism for VLIW and superscalar processors. A serious challenge with compiler-controlled speculative execution is to efficiently handle exceptions for speculative instructions. In this article, a set of architectural features and compile-time scheduling support collectively referred to as sentinel scheduling is introduced. Sentinel scheduling provides an effective framework for both compiler-controlled speculative executi ...

Keywords: VIIW processor, exception detection, exception recovery, instruction scheduling, instruction-level parallelism, speculative execution, superscalar processor

3 Application-level checkpointing for shared memory programs

Greg Bronevetsky, Daniel Marques, Keshav Pingali, Peter Szwed, Martin Schulz

October 2004 **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems**, Volume 32 , 38 , 39 Issue 5 , 5 , 11

Full text available: pdf(235.77 KB) Additional Information: full citation, abstract, references, index terms

Trends in high-performance computing are making it necessary for long-running

applications to tolerate hardware faults. The most commonly used approach is checkpoint and restart (CPR) - the state of the computation is saved periodically on disk, and when a failure occurs, the computation is restarted from the last saved state. At present, it is the responsibility of the programmer to instrument applications for CPR.Our group is investigating the use of compiler technology to instrument codes to ...

Keywords: checkpointing, fault-tolerance, openMP, shared-memory programs

4 Exploiting dead value information

Milo M. Martin, Amir Roth, Charles N. Fischer

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

Publisher Site

We describe Dead Value Information (DVI) and introduce three new optimizations which exploit it. DVI provides assertions that certain register values are dead, meaning they will not be read before being overwritten. The processor can use DVI to track dead registers and dynamically eliminate unnecessary save and restore instructions from the execution stream at procedure calls and context switches. Our results indicate that dynamic saves and restore instances can be reduced by 46% for procedure c ...

5 Hardware-managed register allocation for embedded processors

Xiaotong Zhuang, Tao Zhang, Santosh Pande

June 2004 **ACM SIGPLAN Notices , Proceedings of the 2004 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools**, Volume 39 Issue 7

Full text available: pdf(265.98 KB) Additional Information: full citation, abstract, references, index terms

Most modern processors (either embedded or general purpose) contain higher number of physical registers than those exposed in the ISA. Due to a variety of reasons, this phenomenon is likely to continue especially on embedded systems where encoding space is very limited. Saving the encoding space leads to lower power consumption in the I-cache; on the other hand, harnessing more physical registers saves power in the memory subsystem and reduces latency as well. These design decisions however resu ...

Keywords: architected registers, embedded systems, physical registers, power consumption, register allocation

6 Superscalar architectures: Reducing the complexity of the register file in dynamic superscalar processors

Rajeev Balasubramonian, Sandhya Dwarkadas, David H. Albonesi

December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings

Publisher Site

Dynamic superscalar processors execute multiple instructions out-of-order by looking for independent operations within a large window. The number of physical registers within the processor has a direct impact on the size of this window as most in-flight instructions require a new physical register at dispatch. A large multi-ported register file helps improve the instruction-level parallelism (ILP), but may have a detrimental effect on clock speed, especially in future wire-limited technologies. ...

7 Improving storage system availability with D-GRAID

Muthian Sivathanu, Vijayan Prabhakaran, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau
May 2005 **ACM Transactions on Storage (TOS)**, Volume 1 Issue 2

Full text available: pdf(700.30 KB) Additional Information: full citation, abstract, references, index terms

We present the design, implementation, and evaluation of D-GRAID, a gracefully degrading and quickly recovering RAID storage array. D-GRAID ensures that most files within the file system remain available even when an unexpectedly high number of faults occur. D-GRAID achieves high availability through aggressive replication of semantically critical data, and fault-isolated placement of logically related data. D-GRAID also recovers from failures quickly, restoring only live file system data to a h ...

Keywords: Block-based storage, Disk array, RAID, fault isolation, file systems, smart disks

8 Static single assignment form for machine code
Allen Leung, Lal George
May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation**, Volume 34 Issue 5

Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, citings, index terms

Static Single Assignment (SSA) is an effective intermediate representation in optimizing compilers. However, traditional SSA form and optimizations are not applicable to programs represented as native machine instructions because the use of dedicated registers imposed by calling conventions, the runtime system, and target architecture must be made explicit. We present a simple scheme for converting between programs in machine code and in SSA, such that references to dedicated physical registers ...

9 StackThreads/MP: integrating futures into calling standards
Kenjiro Taura, Kunio Tabata, Akinori Yonezawa
May 1999 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 34 Issue 8

Full text available: pdf(1.58 MB) Additional Information: full citation, abstract, references, citings, index terms

An implementation scheme of fine-grain multithreading that needs no changes to current calling standards for sequential languages and modest extensions to sequential compilers is described. Like previous similar systems, it performs an asynchronous call as if it were an ordinary procedure call, and detaches the callee from the caller when the callee suspends or either of them migrates to another processor. Unlike previous similar systems, it detaches and connects arbitrary frames generated by of ...

10 Inferring annotated types for inter-procedural register allocation with constructor flattening
Torben Amtoft, Robert Muller
January 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN international workshop on Types in languages design and implementation**, Volume 38 Issue 3

Full text available: pdf(268.82 KB) Additional Information: full citation, abstract, references, index terms

We introduce an annotated type system for a compiler intermediate language. The type system is designed to support inter-procedural register allocation and the representation of tuples and variants directly in the register file. We present an algorithm that generates constraints for assigning annotations, and prove its soundness with respect to the type system.

Keywords: certifying compilers, defunctionalization, effects, register allocation, type systems

11 A dynamic multithreading processor
Haitham Akkary, Michael A. Driscoll
November 1998 **Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(2.67 MB) Additional Information: full citation, references, citings, index terms

12 Register connection: a new approach to adding registers into instruction set architectures
Tokuzo Kiyohara, Scott Mahlke, William Chen, Roger Bringmann, Richard Hank, Sadun Anik, Wen-Mei Hwu
May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture**, Volume 21 Issue 2
Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references, citings, index terms

Code optimization and scheduling for superscalar and superpipelined processors often increase the register requirement of programs. For existing instruction sets with a small to moderate number of registers, this increased register requirement can be a factor that limits the effectivess of the compiler. In this paper, we introduce a new architectural method for adding a set of extended registers into an architecture. Using a novel concept of connection, this method allows the data stored in ...

13 Graph coloring register allocation for processors with multi-register operands
Brian R. Nickerson
June 1990 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1990 conference on Programming language design and implementation**, Volume 25 Issue 6
Full text available: pdf(1.41 MB) Additional Information: full citation, abstract, references, citings, index terms

Though graph coloring algorithms have been shown to work well when applied to register allocation problems, the technique has not been generalized for processor architectures in which some instructions refer to individual operands that are comprised of multiple registers. This paper presents a suitable generalization.

14 Fortran 8X draft
Loren P. Meissner
December 1989 **ACM SIGPLAN Fortran Forum**, Volume 8 Issue 4
Full text available: pdf(21.36 MB) Additional Information: full citation, abstract, index terms

Standard Programming Language Fortran. This standard specifies the form and establishes the interpretation of programs expressed in the Fortran language. It consists of the specification of the language Fortran. No subsets are specified in this standard. The previous standard, commonly known as "FORTRAN 77", is entirely contained within this standard, known as "Fortran 8x". Therefore, any standard-conforming FORTRAN 77 program is standard conforming under this standard. New features can b ...

15 Parallel execution of prolog programs: a survey
Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo
July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4

Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

16 NanoFabrics: spatial computing using molecular electronics

Seth Copen Goldstein, Mihai Budiu

May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture**, Volume 29 Issue 2

Full text available: pdf(996.26 KB) Additional Information: full citation, abstract, references, citings, index terms

The continuation of the remarkable exponential increases in processing power over the recent past faces imminent challenges due in part to the physics of deep-submicron CMOS devices and the costs of both chip masks and future fabrication plants. A promising solution to these problems is offered by an alternative to CMOS-based computing, chemically assembled electronic nanotechnology (CAEN).

In this paper we outline how CAEN-based computing can become a reality. We briefly describe rec ...

17 Integrating segmentation and paging protection for safe, efficient and transparent software extensions

Tzi-cker Chiueh, Ganesh Venkitachalam, Prashant Pradhan

December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles**, Volume 33 Issue 5

Full text available: pdf(1.54 MB) Additional Information: full citation, abstract, references, citings, index terms

The trend towards extensible software architectures and component-based software development demands safe, efficient, and easy-to-use extension mechanisms to enforce protection boundaries among software modules residing in the same address space. This paper describes the design, implementation, and evaluation of a novel intra-address space protection mechanism called *Palladium,* which exploits the segmentation and paging hardware in the Intel X86 architecture and efficiently supports safe ...

18 Technical reports

SIGACT News Staff

January 1980 **ACM SIGACT News**, Volume 12 Issue 1

Full text available: pdf(5.28 MB) Additional Information: full citation

19 Empirical evaluation of some features of instruction set processor architectures

Åmund Lunde

March 1977 **Communications of the ACM**, Volume 20 Issue 3

Full text available: Additional Information:

pdf(1.10 MB) full citation, abstract, references, citings

This paper presents methods for empirical evaluation of features of Instruction Set Processors (ISPs). ISP features are evaluated in terms of the time used or saved by having or not having the feature. The methods are based on analysis of traces of program executions. The concept of a register life is introduced, and used to answer questions like: How many registers are used simultaneously? How many would be sufficient all of the time? Most of the time? What would the overhead be if the num ...

Keywords: computer architecture, execution time, instruction sets, instruction tracing, opcode utilization, program behavior, register structures, register utilization, simultaneous register lives

20 Data-Driven and Demand-Driven Computer Architecture

Philip C. Treleaven, David R. Brownbridge, Richard P. Hopkins

January 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 1

Full text available: pdf(4.14 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Results for **"(register and live and physical and logical <in>metadata)"**
Your search matched **27** of **1166705** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» View Session History
» New Search

Modify Search

(register and live and physical and logical <in>metadata)

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Select Article Information

1. **Scalable Coherent Interface**
Alnaes, K.; Kristiansen, E.H.; Gustavson, D.B.; James, D.V.;
CompEuro '90. Proceedings of the 1990 IEEE International Conference on Computer S
Software Engineering
8-10 May 1990 Page(s):446 - 453
AbstractPlus | Full Text: PDF(696 KB) IEEE CNF

2. **IEEE standard for Futurebus+/VME64 Bridge**
IEEE Std 1014.1-1994
17 May 1995
AbstractPlus | Full Text: PDF(4920 KB) IEEE STD

3. **IEEE standard for futurebus+ - physical layer and profile specification**
IEEE Std 896.2-1991
24 April 1992
AbstractPlus | Full Text: PDF(11324 KB) IEEE STD

4. **Helper threads via virtual multithreading**
Wang, P.H.; Collins, J.D.; Dongkeun Kim; Greene, B.; Kai-Ming Chan; Yunus, A.B.; Sy
S.F.; Shen, J.P.; Hong Wang;
Micro, IEEE
Volume 24, Issue 6, Nov.-Dec. 2004 Page(s):74 - 82
AbstractPlus | Full Text: PDF(184 KB) IEEE JNL

5. **Federal law and individual activities**
Walter, C.; Richards, E.P.;
Engineering in Medicine and Biology Magazine, IEEE
Volume 13, Issue 4, Aug.-Sept. 1994 Page(s):544 - 546, 550
AbstractPlus | Full Text: PDF(500 KB) IEEE JNL

6. **Should existing pressure vessel regulations apply to gas filled electrical equipm
engineer's viewpoint**
Rishworth, A.B.;
Energy Conversion, IEEE Transactions on
Volume 6, Issue 2, June 1991 Page(s):259 - 266
AbstractPlus | Full Text: PDF(652 KB) IEEE JNL

7. **Metropolitan area mobile services to support virtual groups**
Walther, U.; Fischer, S.;
Mobile Computing, IEEE Transactions on
Volume 1, Issue 2, Apr-Jun 2002 Page(s):96 - 110
AbstractPlus | References | Full Text: PDF(1166 KB) IEEE JNL

8. **Internet access to a home area network**
Saif, U.; Gordon, D.; Greaves, D.;
Internet Computing, IEEE
Volume 5, Issue 1, Jan.-Feb. 2001 Page(s):54 - 63
AbstractPlus | References | Full Text: PDF(284 KB) IEEE JNL

9. **DataSpace: querying and monitoring deeply networked collections in physical sp**
Imielinski, T.; Goel, S.;
Personal Communications, IEEE [see also IEEE Wireless Communications]
Volume 7, Issue 5, Oct. 2000 Page(s):4 - 9
AbstractPlus | Full Text: PDF(640 KB) IEEE JNL

10. **Biomedical electronics-update**
Lindberg, D.A.B.;
Proceedings of the IEEE
Volume 88, Issue 4, April 2000 Page(s):590 - 592
AbstractPlus | References | Full Text: PDF(40 KB) IEEE JNL

11. **TOPS: an architecture for telephony over packet networks**
Anerousis, N.; Gopalakrishnan, R.; Kalmanek, C.R.; Kaplan, A.E.; Marshall, W.T.; Mish P.Z.; Ramakrishanan, K.K.; Sreenan, C.J.;
Selected Areas in Communications, IEEE Journal on
Volume 17, Issue 1, Jan. 1999 Page(s):91 - 108
AbstractPlus | References | Full Text: PDF(440 KB) IEEE JNL

12. **F.C. Williams**
Cunningham, M.J.;
Engineering Science and Education Journal
Volume 3, Issue 2, April 1994 Page(s):55 - 64
AbstractPlus | Full Text: PDF(824 KB) IEE JNL

13. **Simulation model of the telemedicine program**
Lach, J.M.; Vazquez, R.M.;
Simulation Conference, 2004. Proceedings of the 2004 Winter
Volume 2, 5-8 Dec. 2004 Page(s):2012 - 2017 vol.2
AbstractPlus | Full Text: PDF(546 KB) IEEE CNF

14. **Towards a model for ubiquitous and mobile computing**
Manzoni, S.; Nunnari, F.; Vizzari, G.;
Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004. WET ICE 20 International Workshops on
14-16 June 2004 Page(s):423 - 428
AbstractPlus | Full Text: PDF(152 KB) IEEE CNF

15. **An adaptive protocol for efficient support of range queries in DHT-based system**
Gao, J.; Steenkiste, P.;
Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conf
5-8 Oct. 2004 Page(s):239 - 250
AbstractPlus | Full Text: PDF(521 KB) IEEE CNF

16. **Demand side management in private homes by using LonWorks**
Palensky, P.; Dietrich, D.; Posta, R.; Reiter, H.;
Factory Communication Systems, 1997. Proceedings.1997 IEEE International Worksh
1-3 Oct. 1997 Page(s):341 - 347
AbstractPlus | Full Text: PDF(456 KB) IEEE CNF

17. **Design of a persistent operating system kernel**
Kemikli, E.; Erdogan, N.;
Electrotechnical Conference, 1998. MELECON 98., 9th Mediterranean
Volume 2, 18-20 May 1998 Page(s):1304 - 1307 vol.2
AbstractPlus | Full Text: PDF(364 KB) IEEE CNF

18. **CUMULVS: extending a generic steering and visualization middleware for applic tolerance**
Papadopoulos, P.M.; Kohl, J.A.; Semeraro, B.D.;
System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conferen
Volume 7, 6-9 Jan. 1998 Page(s):127 - 136 vol.7
AbstractPlus | Full Text: PDF(900 KB) IEEE CNF

19. **Protecting resources with resource control lists**
Miyoshi, A.; Rajkumar, R.;
Real-Time Technology and Applications Symposium, 2001. Proceedings. Seventh IEE
30 May-1 June 2001 Page(s):85 - 94
AbstractPlus | Full Text: PDF(952 KB) IEEE CNF

20. **User interfaces for network services: what, from where, and how**
Ponnekanti, S.R.; Robles, L.A.; Fox, A.;
Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Worksh
20-21 June 2002 Page(s):138 - 147
AbstractPlus | Full Text: PDF(580 KB) IEEE CNF

21. **Distributed applications for collaborative augmented reality**
Schmalstieg, D.; Hesina, G.;
Virtual Reality, 2002. Proceedings. IEEE
24-28 March 2002 Page(s):59 - 66
AbstractPlus | Full Text: PDF(3026 KB) IEEE CNF

22. **A coarse-grain phased logic CPU**
Reese, R.B.; Thornton, M.A.; Traver, C.;
Asynchronous Circuits and Systems, 2003. Proceedings. Ninth International Symposiu
12-15 May 2003 Page(s):2 - 13
AbstractPlus | Full Text: PDF(395 KB) IEEE CNF

23. **Multimedia wireless interactive and collaborative telecom services**
Choukair, Z.; Takizawa, M.;
Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Cor
23-24 March 2004 Page(s):150 - 155
AbstractPlus | Full Text: PDF(358 KB) IEEE CNF

24. **Multi-protocol profiles to support user mobility across network technologies**
Haase, O.; Ming Xiong; Murakami, K.;
Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on
2004 Page(s):100 - 105
AbstractPlus | Full Text: PDF(306 KB) IEEE CNF

25. **Part 3: Carrier sense multiple access with collision detect on (CSMA/CD) access physical layer specifications**

IEEE Std 802.3, 2000 Edition
2000 Page(s):i - 1515
AbstractPlus | Full Text: PDF(19532 KB) IEEE STD

Terms used **register** and **live** and **physical** and **logical** and **stack** Found **46,717** of **866,341**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The Digital Library

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Hardware-managed register allocation for embedded processors

Xiaotong Zhuang, Tao Zhang, Santosh Pande

June 2004 **ACM SIGPLAN Notices , Proceedings of the 2004 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools**, Volume 39 Issue 7

Full text available: pdf(265.98 KB) Additional Information: full citation, abstract, references, index terms

Most modern processors (either embedded or general purpose) contain higher number of physical registers than those exposed in the ISA. Due to a variety of reasons, this phenomenon is likely to continue especially on embedded systems where encoding space is very limited. Saving the encoding space leads to lower power consumption in the I-cache; on the other hand, harnessing more physical registers saves power in the memory subsystem and reduces latency as well. These design decisions however resu ...

Keywords: architected registers, embedded systems, physical registers, power consumption, register allocation

2 Physical Register Inlining

Mikko H. Lipasti, Brian R. Mestan, Erika Gunadi

March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st annual international symposium on Computer architecture - Volume 00**, Volume 32 Issue 2

Full text available: pdf(273.94 KB) Additional Information: full citation, abstract

Physical register access time increases the delaybetween scheduling and execution in modern out-of-orderprocessors. As the number of physical registers increases,this delay grows, forcing designers to employ register fileswith multicycle access. This paper advocates more efficientutilization of a fewer number of physical registers in orderto reduce the access time of the physical register file. Registervalues with few significant bits are stored in the renamemap using physical register inlining, ...

3 Exploiting Value Locality in Physical Register Files

Saisanthosh Balakrishnan, Gurindar S. Sohi

December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture**

Full text available: pdf(194.25 KB) Additional Information: full citation, abstract, index terms

The physical register file is an important component of adynamically-scheduled processor. Increasing the amount of parallelismplaces increasing demands on the physical register

file,calling for alternative file organization and management strategies.This paper considers the use of value locality to optimize theoperation of physical register files.We present empirical data showing that: (i) the value producedby an instruction is often the same as a value produced by anotherrecently executed instr ...

4 Safe Class and Data Evolution in Large and Long-Lived Java[tm] Applications
Mikhail Dmitriev
August 2001 Technical Report, Sun Microsystems, Inc.

Full text available: pdf(876.82 KB) Additional Information: full citation, abstract

There is a growing class of applications implemented in object-oriented languages that are large and complex, that exploit object persistence, and need to run uninterrupted for long periods of time. Development and maintenance of such applications can present challenges in the following interrelated areas: consistent and scalable evolution of persistent data and code, optimal build management, and runtime changes to applications. The research presented in this thesis addresses the above issues. ...

5 Exploiting dead value information
Milo M. Martin, Amir Roth, Charles N. Fischer
December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

We describe Dead Value Information (DVI) and introduce three new optimizations which exploit it. DVI provides assertions that certain register values are dead, meaning they will not be read before being overwritten. The processor can use DVI to track dead registers and dynamically eliminate unnecessary save and restore instructions from the execution stream at procedure calls and context switches. Our results indicate that dynamic saves and restore instances can be reduced by 46% for procedure c ...

6 Improving superscalar instruction dispatch and issue by exploiting dynamic code sequences
Sriram Vajapeyam, Tulika Mitra
May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2

Full text available: pdf(1.76 MB) Additional Information: full citation, abstract, references, citings, index terms

Superscalar processors currently have the potential to fetch multiple basic blocks per cycle by employing one of several recently proposed instruction fetch mechanisms. However, this increased fetch bandwidth cannot be exploited unless pipeline stages further downstream correspondingly improve. In particular, register renaming a large number of instructions per cycle is difficult. A large instruction window, needed to receive multiple basic blocks per cycle, will slow down dependence resolution ...

7 Superscalar architectures: Reducing the complexity of the register file in dynamic superscalar processors
Rajeev Balasubramonian, Sandhya Dwarkadas, David H. Albonesi
December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings
Publisher Site

Dynamic superscalar processors execute multiple instructions out-of-order by looking for independent operations within a large window. The number of physical registers within the

processor has a direct impact on the size of this window as most in-flight instructions require a new physical register at dispatch. A large multi-ported register file helps improve the instruction-level parallelism (ILP), but may have a detrimental effect on clock speed, especially in future wire-limited technologies. ...

8 Computing curricula 2001

September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available: pdf(613.63 KB) html(2.78 KB) Additional Information: full citation, references, citings, index terms

9 Static single assignment form for machine code

Allen Leung, Lal George

May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation**, Volume 34 Issue 5

Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, citings, index terms

Static Single Assignment (SSA) is an effective intermediate representation in optimizing compilers. However, traditional SSA form and optimizations are not applicable to programs represented as native machine instructions because the use of dedicated registers imposed by calling conventions, the runtime system, and target architecture must be made explicit. We present a simple scheme for converting between programs in machine code and in SSA, such that references to dedicated physical registers ...

10 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

11 StackThreads/MP: integrating futures into calling standards

Kenjiro Taura, Kunio Tabata, Akinori Yonezawa

May 1999 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 34 Issue 8

Full text available: pdf(1.58 MB) Additional Information: full citation, abstract, references, citings, index terms

An implementation scheme of fine-grain multithreading that needs no changes to current calling standards for sequential languages and modest extensions to sequential compilers is described. Like previous similar systems, it performs an asynchronous call as if it were an ordinary procedure call, and detaches the callee from the caller when the callee suspends or either of them migrates to another processor. Unlike previous similar systems, it detaches and connects arbitrary frames generated by of ...

12 Use-Based Register Caching with Decoupled Indexing

J. Adam Butts, Gurindar S. Sohi

March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st**

annual international symposium on Computer architecture - Volume 00, Volume 32 Issue 2

Full text available: pdf(182.25 KB) Additional Information: full citation, abstract

Wide, deep pipelines need many physical registersto hold the results of in-flight instructions. Simultaneously,high clock frequencies prohibit using largeregister files and bypass networks without a significantperformance penalty. Previously proposed techniquesusing register caching to reduce this penalty sufferfrom several problems including poor insertion andreplacement decisions and the need for a fully-associativecache for good performance. We present novelmechanisms for managing and indexin ...

13 Supermachines and Superminds

Eric Steinhart

February 2003 **Minds and Machines**, Volume 13 Issue 1

Full text available: Publisher Site Additional Information: full citation, abstract, references, index terms

If the computational theory of mind is right, then minds are realized by machines. There is an ordered complexity hierarchy of machines. Some finite machines realize finitely complex minds; some Turing machines realize potentially infinitely complex minds. There are many logically possible machines whose powers exceed the Church–Turing limit (e.g. accelerating Turing machines). Some of these supermachines realize superminds. Superminds perform cognitive supertasks. Their thoughts are fo ...

Keywords: complexity, divine mind, infinite computer, infinite mind, supertask

14 First International Workshop on Persistence and Java

Malcolm Atkinson, Mick Jordan

November 1996 Technical Report, Sun Microsystems, Inc.

Full text available: pdf(1.54 MB) Additional Information: full citation, abstract

These proceedings record the First International Workshop on Persistence and Java, which was held in Drymen, Scotland in September 1996. The focus of this workshop was the relationship between the Java languages and long-term data storage, such as databases and orthogonal persistence. There are many approaches being taken, some pragmatic and some guided by design principles. If future application programmers building large and long-lived systems are to be well-supported, it is essential that the ...

15 A dynamic multithreading processor

Haitham Akkary, Michael A. Driscoll

November 1998 **Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(2.67 MB) Additional Information: full citation, references, citings, index terms

16 Proceedings of the Second International Workshop on Persistence and Java

Malcolm Atkinson, Mick Jordan

December 1997 Technical Report, Sun Microsystems, Inc.

Full text available: pdf(1.23 MB) Additional Information: full citation, abstract

These proceedings record the Second International Workshop on Persistence and Java, that was held in Half Moon Bay in the San Francisco Bay Area, in August 1997. The focus of the workshop series is the relationship between the Java platform and longterm storage, such as databases and orthogonal persistence. If future application programmers building large and longlived systems are to be well supported, it is essential that the lessons of existing

research into language and persistence combinatio ...

17 The Multicluster Architecture: Reducing Processor Cycle Time Through Partitioning
Keith I. Farkas, Paul Chow, Norman P. Jouppi, Zvonko Vranesic
October 1999 **International Journal of Parallel Programming**, Volume 27 Issue 5

Full text available: Publisher Site Additional Information: full citation, abstract, references, index terms

The multicluster architecture that we introduce offers a decentralized, dynamically-scheduled architecture, in which the register files, dispatch queue, and functional units of the architecture are distributed across multiple clusters, and each cluster is assigned a subset of the architectural registers. The motivation for the multicluster architecture is to reduce the clock cycle time, relative to a single-cluster architecture with the same number of hardware resources, by reducing the size ...

Keywords: PARTITIONED DYNAMICALLY-SCHEDULED ARCHITECTURE, REGISTER ALLOCATION, STATIC INSTRUCTION SCHEDULING

18 A novel renaming mechanism that boosts software prefetching
Daniel Ortega, Mateo Valero, Eduard Ayguadé
June 2001 **Proceedings of the 15th international conference on Supercomputing**

Full text available: pdf(214.40 KB) Additional Information: full citation, abstract, references, index terms

The detection and correct handling of data and control dependencies constitutes one of the biggest issues to expose ILP in current architectures. The ever increasing memory latencies and working space of programmes are making prefetching techniques crucial for the attainment of sustained high performance. Software prefetching allows the compiler to use information discovered at compile-time to effectively bring needed data before it is used, thus hiding all or part of the latency from main me ...

19 The multicluster architecture: reducing cycle time through partitioning
Keith I. Farkas, Paul Chow, Norman P. Jouppi, Zvonko Vranesic
December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.44 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

The multicluster architecture that we introduce offers a decentralized, dynamically-scheduled architecture, in which the register files, dispatch queue, and functional units of the architecture are distributed across multiple clusters, and each cluster is assigned a subset of the architectural registers. The motivation for the multicluster architecture is to reduce the clock cycle time, relative to a single-cluster architecture with the same number of hardware resources, by reducing the size and ...

Keywords: decentralized architecture, partitioned architecture, static instruction scheduling, register allocation

20 Epochs
Jon A. Solworth
January 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 14 Issue 1

Full text available: pdf(1.68 MB) Additional Information: full citation, abstract, references, index terms, review

To date, the implementation of message passing languages has required hte

communications variables (sometimes called ports) either to be limited to the number of physical communications registers in the machine or to be mapped to memory. Neither solution is satisfactory. Limiting the number of variables decreases modularity and efficiency of parallel programs. Mapping variables to memory increases the cost of communications and the granularity of parallelism. We present here a new programmi ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next